

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Peningkatan penggunaan jaringan saat ini seperti pemakaian perangkat, virtualisasi, *big data*, ataupun komputasi awan (*cloud computing*) menyebabkan perubahan pola trafik jaringan dan semakin kompleksnya suatu jaringan. *Software Defined Network* (SDN) dapat memenuhi permasalahan yang ada, karena jaringan diaplikasikan sesuai kebutuhan, memenuhi kompleksitas jaringan dan membangun paradigma baru untuk merevolusi jaringan yang ada saat ini, dibandingkan menggunakan jaringan tradisional [1]. SDN merupakan arsitektur jaringan dengan konsep memisahkan antara *control plane* dan *data plane*, pemisahan dilakukan agar pemrograman, fleksibilitas, dan skalabilitas dapat dilakukan secara terpusat [2]. Fungsi *control plane* adalah melakukan aktivitas yang diperlukan untuk menjalankan fungsi *data plane*, melibatkan paket dari pengguna dan fungsi *data plane* adalah melakukan aktivitas yang melibatkan hasil pemrosesan paket yang dikirim oleh pengguna [3]. Didalam *control plane* terdapat *controller* untuk mengelola dan mengatur jalur dari dan untuk *data plane*. *Controller* yang dipakai dalam SDN sangat banyak, salah satunya adalah RYU. RYU adalah *controller* SDN yang dirancang agar *administrator* mudah untuk mengelola dan menyesuaikan bagaimana cara penanganan lalu lintas jaringan [4]. Diantara *control plane* dan *data plane* pada SDN, terdapat protokol yang menghubungkannya yaitu *OpenFlow*. *OpenFlow* adalah protokol untuk pengaturan jalur paket ketika melalui sebuah *switch* [5]. Agar SDN bisa disimulasikan secara virtual, SDN dijalankan pada *Mininet*, *Mininet* adalah pembuat jaringan *virtual* untuk arsitektur SDN [6].

Penelitian ini memfokuskan arsitektur SDN dengan menggunakan *Internet Protocol version 6* (IPv6). IPv6 adalah protokol pengembangan dari IPv4, yang diciptakan dan dikembangkan karena keterbatasan ruang alamat di IPv4 yang hanya memakai alamat 32 bit. IPv6 menggunakan alamat 128 bit yang memungkinkan 2128 kali lebih banyak dari IPv4 [7]. Dalam IPv6, untuk mengetahui perangkat-perangkat yang terhubung dalam suatu jaringan, menggunakan fungsi *Neighbor Discovery Protocol* (NDP), *Internet Control Message Protocol version 6* (ICMPv6)

dan *multicast*. *Multicast* merupakan teknik untuk mengirimkan paket dari satu titik ke titik-titik khusus. ICMPv6 merupakan pelaporan kesalahan dalam pengolahan suatu paket atau pesan, dan salah satu contoh tipe pesan ICMPv6 adalah NDP. NDP merupakan proses untuk membangun dan memelihara koneksi antar sesama IPv6 dalam satu jaringan. Di dalam NDP terdapat lima jenis pesan yang melakukan fungsi terkait dengan NDP yaitu *Router Solicitation* (RS), *Router Advertisement* (RA), *Neighbor Solicitation* (NS), *Neighbor Advertisement* (NA), dan *Redirect* [8].

Beberapa rujukan utama yang menjadi acuan dalam membuat penelitian ini yaitu pertama penelitian Fauzi Dwi, dkk dengan judul *Controller Based Proxy for Handling NDP in OpenFlow Network* [9]. Kedua penelitian Roberto Bifulco, dkk dengan judul *Improving SDN with InSPired Switches* [10]. Ketiga penelitian Talal Alharbi, dkk dengan judul *SProxy ARP - Efficient ARP Handling in SDN*. [11]. Persamaan ketiga penelitian ini memungkinkan *controller* melepas sebagian tugasnya namun tetap mempertahankan kontrol penuh atas jaringannya sesuai standar OpenFlow dan SDN, dan untuk mengurangi beban kerja CPU. Perbedaan ketiga penelitian ini meliputi, penelitian pertama menggunakan pendekatan *Reactive Proxy* di IPv6, penelitian kedua API yang dipasang di *Switch* di IPv4, dan penelitian ketiga menggunakan pendekatan baru, *Switch-based Proxy ARP* (SProxy ARP) di IPv4.

Penelitian ini bertujuan untuk membuat pendekatan baru seperti penelitian ketiga. Dan penelitian ini merupakan pengembangan penelitian pertama dan kedua yaitu pendekatan *Reactive Proxy*, maksudnya jika ada permintaan paket NS dari *host*, permintaan tersebut dikirim ke *controller*. *Controller* akan mempertahankan pemetaan alamat MAC, DPID, dan *port* nya, dan jika sudah ada pemetaan, dapat langsung membuat pesan balasan yang sesuai dengan pencocokan. Dari penelitian-penelitian diatas, penulis mengembangkan aturan *proxy* yang dapat dikirim ke dalam *switch* jaringan SDN tanpa merubah prinsip OpenFlow dan SDN di IPv6, yang diberi nama pendekatan *Semi Reactive SProxy*. Pendekatan tersebut dibandingkan dengan pendekatan *Reactive Proxy*, dan mengukur dengan skenario penambahan beban trafik UDP dan berbagai tingkat pengiriman paket NS. Hasilnya yaitu *NDP Response Time* yang didapatkan dari waktu diterimanya NA dikurangi waktu dikirimnya NS, *packet loss*, dan persentase CPU *controller*.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang, maka dapat dirumuskan beberapa permasalahan, antara lain:

- a. Bagaimana implementasi pendekatan *Reactive Proxy* dan pendekatan *Semi Reactive SProxy* dalam IPv6 di arsitektur SDN menggunakan *controller* RYU yang disimulasikan dalam *Mininet*?
- b. Bagaimana hasil perbandingan setelah dilakukan pendekatan *Semi Reactive SProxy* dan pendekatan *Reactive Proxy* dengan melihat dari hasil pengujian?

## 1.3. Tujuan Penelitian

Tujuan dari penelitian tugas akhir ini adalah implementasi pendekatan *Semi Reactive SProxy* untuk mengurangi *NDP Response Time* dalam proses NDP, hasilnya mengurangi beban kerja *controller* untuk meningkatkan skalabilitas jaringan tanpa merubah standar dari SDN maupun OpenFlow yang ada dan membandingkannya dengan pendekatan *Reactive Proxy*.

## 1.4. Cakupan Masalah

Adapun cakupan masalah dalam proposal tugas akhir ini, antara lain:

- a. Protokol *internet* yang digunakan adalah *Internet Protocol version 6* (IPv6) dan hanya membahas *Neighbor Solicitation* dan *Neighbor Advertisement*.
- b. Menggunakan *Link-Local Address* atau subnet yang sama yaitu *fe80::/64*.
- c. Arsitektur jaringan menggunakan arsitektur *Software Defined Network*.
- d. Sistem operasi yang digunakan adalah *Linux Ubuntu 16.04 LTS*.
- e. Simulator jaringan SDN yang digunakan adalah *Mininet* versi 2.2.2.
- f. Topologi yang diimplementasikan dalam *Mininet* adalah topologi *tree*.
- g. *Controller* yang dipakai adalah *RYU Network Operating Sistem*.
- h. Protokol yang digunakan adalah *OpenFlow* versi 1.3.
- i. Penanganan NDP dengan *Semi Reactive SProxy* dan *Reactive Proxy*.
- j. Membuat manipulasi paket menggunakan *Scapy* versi 2.3.3.
- k. Mengirimkan paket menggunakan *Tcpreplay* versi 4.0.
- l. Penambahan beban trafik UDP menggunakan *Iperf* versi 3.1.
- m. Tidak membahas mengenai keamanan jaringan.

### **1.5. Pembuatan Laporan**

Melakukan penyusunan laporan hasil implementasi dan pengujian serta analisa dari serangkaian penelitian yang sudah dilakukan. Penulis mengambil kesimpulan dari hasil pengujian dan analisa yang kemudian dituliskan kedalam bentuk laporan tugas akhir.

### **1.6. Sistematika Penulisan**

Bagian isi dari tugas akhir ini disusun kedalam beberapa bab-bab, antara lain:

#### **BAB I PENDAHULUAN**

Bagian ini berisi ulasan permasalahan yang ada dalam jaringan arsitektur tradisional dan SDN yang menjadi dasar perlunya dilakukan pengembangan penelitian disertai pendekatannya yang dipakai. Tersusun dari latar belakang, rumusan masalah, tujuan penelitian, cakupan masalah, pembuatan laporan dan sistematika penulisan.

#### **BAB II TINJAUAN PUSTAKA**

Bagian ini membahas dasar-dasar studi literatur yang digunakan untuk menjadi landasan dari topik penelitian dalam tugas akhir ini yang berjudul “*Semi Reactive Approach for Network Discovery Protocol-Switch Based Proxy in Software Defined Network*”.

#### **BAB III METODE PENELITIAN**

Bagian ini berisi tentang metode penelitian yang dilakukan yaitu identifikasi masalah, studi literatur, perancangan sistem terkait rancangan topologi jaringan, rancangan pendekatan *Reactive Proxy*, dan rancangan pendekatan *Semi Reactive SProxy*, pembangunan sistem terkait pembangunan pendekatan *Reactive Proxy* dan pembangunan pendekatan *Semi Reactive SProxy*, implementasi dan skenario pengujian sistem yang dipakai dalam topik penelitian tugas akhir ini. Skenario pengujiannya adalah sebagai berikut:

1. Pengujian pertama adalah mengamati perbandingan pendekatan *Semi Reactive SProxy* dan *Reactive Proxy* dalam kondisi jaringan yang normal dengan mengirimkan satu paket NS per detik selama tiga menit dan mengamati hasil *NDP Response Time* (ms) dalam kondisi normal.

2. Pengujian kedua adalah mengamati perbandingan pendekatan *Reactive Proxy* dan *Semi Reactive SProxy* dengan penambahan trafik UDP dalam jaringan dengan berbagai tingkat trafik UDP yaitu 2, 10, 50, 100, dan 500 (Mbps) dengan mengirimkan satu paket NS per detik selama tiga menit dan mengamati hasil NDP *Response Time* (ms).
3. Pengujian ketiga adalah mengamati perbandingan pendekatan *Reactive Proxy* dan *Semi Reactive SProxy* dalam kondisi jaringan normal dari berbagai tingkat pengiriman paket NS yaitu 1000, 2000, 3000, 4000, dan 5000 (paket per detik) selama satu menit dengan total paket yaitu 60000, 120000, 180000, 240000, dan 300000, dan mengamati hasil NDP *Response Time* (ms).
4. Pengujian keempat adalah mengamati perbandingan pendekatan *Reactive Proxy* dan *Semi Reactive SProxy* dalam kondisi jaringan normal dari berbagai tingkat pengiriman paket NS yaitu 1000, 2000, 3000, 4000, 5000 (paket per detik) selama satu menit dengan total paket yaitu 60000, 120000, 180000, 240000, dan 300000, dan mengamati jumlah *packet loss*.
5. Pengujian kelima adalah mengamati perbandingan pendekatan *Semi Reactive SProxy* dan *Reactive Proxy* dalam kondisi jaringan normal dari berbagai tingkat pengiriman paket NS yaitu 1000, 2000, 3000, 4000, 5000 (paket per detik) selama satu menit dengan total paket yaitu 60000, 120000, 180000, 240000, dan 300000, dan mengamati hasil persentase CPU *controller*.

#### **BAB IV IMPLEMENTASI DAN HASIL PENGUJIAN**

Bagian ini membahas cara dan hasil bagaimana mengimplementasikan pendekatan *Reactive Proxy* dan pendekatan *Semi Reactive SProxy* dalam jaringan SDN dalam *Mininet* dengan topologi *tree* dan memakai *controller* RYU.

#### **BAB V PENUTUP**

Bagian ini berisi kesimpulan yang menjadi ikhtisar dari serangkaian penelitian yang sudah dilakukan dan saran untuk penelitian terkait berikutnya yang akan datang.